TITLE:

An Algorithm Which Generates the Linear Extensions of a d-Complete Poset with Uniform Probability (Theoretical Computer Science and Its Applications)

AUTHOR(S):

仲田, 研登; 岡村, 修志

CITATION:

仲田, 研登 ...[et al]. An Algorithm Which Generates the Linear Extensions of a d-Complete Poset with Uniform Probability (Theoretical Computer Science and Its Applications). 数理解析研究所講究録 2009, 1649: 1-8

ISSUE DATE:

2009-05

URL:

http://hdl.handle.net/2433/140763

RIGHT:

京都大学学術情報リポジトリ
KURENAI 紅
Kyoto University Research Information Repository

数理解析研究所講究録
第1649巻 2009年 1-8

An Algorithm Which Generates the Linear Extensions of a d-Complete Poset with Uniform Probability

By

KENTO NAKADA* and SHUJI OKAMURA**

§1. 研究の背景, モチベーション, 既存の結果と我々の結果の比較

まず, 主結果を述べる:

主結果 一般化された GNW-algorithm (defined in section 3) は d-complete poset (defined in section 4) の linear extension を等確率に生成する (Theorem 5.1). このアルゴリズムは d-complete poset の linear extension の多項式時間での counting を可能にする (Corollary 5.3).

はじめに, 我々の研究の計算機科学における研究背景やモチベーションを述べ, 既存の結果との比較を行いたい.

その1 任意に与えられた poset の linear extension の総数を求める問題は, G. Brightwell, P. Winkler [2] によって, $\#P$-完全であることが分かっている. また, acyclic graph の linear extension は, この acyclic graph から自然に定まる poset の linear extension と一致する. したがって, 任意に与えられた acyclic graph の linear extension の総数を求める問題も $\#P$-完全である.

このような状況においては, linear extension の総数が (多項式時間で) 求められる出来るだけ大きなクラスを獲得することに一定の意味がある. 我々の結果は今までは知られていなかった (比較的) 大きなクラスを与える. (d-complete poset のなすクラスは, linear extension の総数が多項式時間で求められる現状で最も大きなクラスである.)
もう一つ, 特筆すべき点は, 我々の counting が確率生成アルゴリズムに基づいていることにある.

2000 Mathematics Subject Classification(s): Primary 05E10, Scondary 68W20
Key Words: *d-complete poset, linear extension, uniform generation, counting*
*Research Institute for Mathematical Sciences, Kyoto University, Kyoto 606-8502, JAPAN,
Partially supported by GCOE, Kyoto University
e-mail: nakada@kurims.kyoto-u.ac.jp
**Osaka Prefectural College of Technology,
e-mail: okamura@ipc.osaka-pct.ac.jp

その2 poset の linear extension を確率的に生成するアルゴリズムの研究は数多くあるが, A. Karzanov-L. Khachiyan [6] が先駆的である. [6] では任意の poset の linear extension を一つあたり多項式時間で確率的に生成するアルゴリズムが構成された. 後に, このアルゴリズムは linear extension を一つあたり $O(d^3 \log d)$-time で確率的に生成していることが, D. B. Wilson によって示されている (d は頂点の個数). その後, R. Bubley-M. Dyer [1], M. Huber [5], によって, (アルゴリズムは違うが, やはり) 一つあたり $O(d^3 \log d)$-time の確率生成アルゴリズムが構成されている. 特に [5] のアルゴリズムは秀逸で, すべての linear extension を正確に等確率で生成する.

我々のアルゴリズムは poset の範囲を制限する代わりに, 一つあたり $O(d^2)$-time のアルゴリズムになっており, しかも, 一つあたりの生成確率は 定量的に等確率[1] である (Theorem 5.1) (この事実から, counting が可能となる).[2]

先行研究の紹介と結果の比較 (L = linear extension の個数, d = 頂点の個数)				
対象	一つあたりの生成確率	counting の機能	一つあたりの生成時間	論文
一般の poset	だいたい $1/L$	×	多項式時間 ($O(d^3 \log d)$)	A. Karzanov-L. Khachiyan [6] (by D. B. Wilson)
一般の poset	だいたい $1/L$	×	$O(d^3 \log d)$	R. Bubley-M. Dyer [1]
一般の poset	正確に $1/L$ 非定量的	△	$O(d^3 \log d)$	M. Huber [5]
d-complete poset	正確に $1/L$ 定量的	○	$O(d^2)$	K. Nakada-S. Okamura [11]

我々の研究の直接の先行研究は以下の研究である:

その3 C.Greene-A.Nijenhuis-H.S.Wilf は 1979 年の論文 [4] で, Young diagram が定める acyclic graph (または poset) の linear extension を等確率に生成する random walk (あるいは Markov chain) に基づくアルゴリズム (GNW-algorithm) を考案した.
また, B. E. Sagan は G-N-W の方法と類似のアルゴリズムを構成し, それを shifted Young diagram の場合に適用して [14], (同様の) 類似の結果を得た.

我々は, この G-N-W の方法と Sagan の方法を同時に一般化するアルゴリズムを構成した. そしてこのアルゴリズムを d-complete poset に適用して, より一般的な結果を得た.

Remark 1. Young diagram や shifted Young diagram (が定める acyclic graph) は d-complete poset である. したがって, 我々の結果は [4] [14] の結果を含む.

---

[1] 等確率であり, しかもその確率が (多項式時間で) 計算できるときに (本稿では) 定量的に等確率と呼ぶ.

[2] [5] の方法では生成確率は等確率だが 非定量的 であり, counting は estimate しか行えない. もし counting も行えるようであれば, [2] の結果と合わせて, $\#P$-問題はすべて決定性テューリングマシンで多項式時間で解ける, ことになってしまう.

§2. 準備

Definition 1. $\Gamma = (V; A, \mathrm{o}, \mathrm{i})$ が *multi-di-graph* とは:

$$\begin{aligned} V &: \text{ a set (an element of } V \text{ is called a vertex of } \Gamma \text{)} \\ A &: \text{ a set (an element of } A \text{ is called an arrow of } \Gamma \text{)} \\ \mathrm{o} : A \longrightarrow V &: \text{ a map (an arrow } a \in A \text{ comes out from } \mathrm{o}(a) \in V \text{)} \\ \mathrm{i} : A \longrightarrow V &: \text{ a map (an arrow } a \in A \text{ goes into } \mathrm{i}(a) \in V \text{)} \end{aligned}$$

のことである.

Definition 2. Γ を multi-di-graph とする. $a \in A(\Gamma)$ とする. $\mathrm{o}(a) = v, \mathrm{i}(a) = u$ のとき,

$$v \xrightarrow{a} u$$

と書く.

Definition 3. infinite v-path とは, arrow の無限列 $(a_1, a_2, a_3, \cdots)$ で,

$$\begin{aligned} &v = \mathrm{o}(a_1), \text{ and} \\ &\mathrm{i}(a_k) = \mathrm{o}(a_{k+1}) \text{ for each } k = 1, 2, 3 \cdots, \end{aligned}$$

を満たすものである.

Definition 4. $v \in V(\Gamma)$ に対して, 集合 $\phi(v)$ を次で定義する:

$$\phi(v) := \left\{ a \in A \,\middle|\, \mathrm{o}(a) = v \right\}.$$

集合 $\phi(v)$ を (つまり v から出る arrow の集合を) v の strict hook と呼ぶ.

Definition 5. $u \in V(\Gamma)$ に対して, 集合 $\psi(u)$ を次で定義する:

$$\psi(u) := \left\{ a \in A(\Gamma) \,\middle|\, \mathrm{i}(a) = u \right\}.$$

集合 $\psi(u)$ を (つまり u に入る arrow の集合を) u の strict cohook と呼ぶ.

今後しばらく, 考えるグラフは以下の性質 (D0),(D1),(D2) を満たすと仮定する[3]:

(D0) $\#V(\Gamma) < \infty$.

(D1) 任意の $v \in V(\Gamma)$ に対して, $\#\phi(v) < \infty$ である.

(D2) 任意の $v \in V(\Gamma)$ に対して, Γ は infinite v-path を含まない.

Remark 2. 条件 (D2) よりグラフ Γ が cycle を持たないことが従う. したがって, $v, u \in V$ に対して, $v > u$ を

$$\exists n \geq 1, \exists a_1, \cdots, a_n \in A \text{ s.t. } v = \mathrm{o}(a_1), \mathrm{i}(a_i) = \mathrm{o}(a_{i+1}) \ (i = 1, \cdots n-1), \mathrm{i}(a_n) = u,$$

と定めれば, これは V 上に半順序を定める. このように定めた $(V; >)$ を Γ に付随する poset と呼ぶ.

[3] このような定義にしているのは, 将来 (D0) を外すことを想定しているからであるが, 要するに, acyclic な graph のことである (ただし multi-arrow は持ちうる).

§3. 一般化された GNW-algorithm の定義

一般化された GNW-algorithm は2段階の procedure からなる.
まず, $\#V(\Gamma) \neq 0$ であるグラフに対して, 次のアルゴリズムを考える:

- Procedure CHW(Γ)

10: Chose an element $v \in V(\Gamma)$ with a probability $\dfrac{1}{\#V(\Gamma)}$

20: **if** $\#\phi(v) = 0$ **then goto** 60

30: Chose an element $a \in \phi(v)$ with a probability $\dfrac{1}{\#\phi(v)}$

40: PUT $v := \mathrm{i}(a)$

50: **goto** 20

60: OUTPUT v

70: **stop**

このアルゴリズムによって $v \in V(\Gamma)$ s.t. $\phi(v) = \varnothing$ が確率的に選ばれる.

Remark 3.
(D0) より, 確率 $\dfrac{1}{\#V(\Gamma)}$ は $V(\Gamma) \neq \varnothing$ である限り定義される.
(D1) より, 確率 $\dfrac{1}{\#\phi(v)}$ は $\phi(v) \neq \varnothing$ である限り定義される.
(D2) より, このアルゴリズムは有限時間 ($O(d)$-time) で停止する.

Remark 4. このアルゴリズムは要するに, Γ の頂点を一様ランダムに選び, そこから random walk を行い, 停まったらその頂点を出力せよ, という意味である.

さらに次のアルゴリズムを考える:

- Procedure GNW(Γ)

10: **if** $\#V(\Gamma) = 0$ **then goto** 50

20: **RUN** Procedure CHW(Γ)(Procedure CHW(Γ) から得られる OUTPUT を v とする)

30: PUT $\Gamma := \Gamma - v$ (Γ における $V(\Gamma) - \{v\}$ による誘導部分グラフ)

40: **goto** 10

50: **stop**

このアルゴリズムによって Γ の頂点列 $\mathcal{B}=(v_1,\cdots,v_d)$ が確率的に選ばれる. この確率を $\mathrm{Prob}_\Gamma(\mathcal{B})$ と書く. ただし, ここで $d=\#V(\Gamma)$.

Remark 5. 定義より, このアルゴリズムは有限時間 ($O(d^2)$-time) で停止する.

Definition 6. Γ の頂点列 $\mathcal{B}=(v_1,\cdots,v_d)$ が Γ の linear extension であるとは次を満たすことである:

$$\text{if } v_p \to v_q, \text{ then we have } p>q, \quad (p,q\in\{1,\cdots,d\}).$$

Γ の linear extension の全体を $\mathcal{L}(\Gamma)$ と書く.

アルゴリズムの定義から直ちに次を得る:

Proposition 3.1. *Procedure GNW(Γ)* が確率的に生成する頂点列 $(v_1,\cdots,v_d)$ は Γ の *linear extension* である.

Procedure GNW(Γ) は $\mathcal{L}(\Gamma)$ の元をランダムに出力するアルゴリズムである.
言い換えれば, Procedure GNW (Γ) は有限集合 $\mathcal{L}(\Gamma)$ 上に確率分布 $\mathrm{Prob}_\Gamma()$ を与える.
もちろん, 一般の Γ では, この確率分布が一様分布になることは到底期待できない.

補足 poset $P=(V;>)$ $(d=\#V)$ の頂点列 $\mathcal{B}=(v_1,\cdots,v_d)$ が P の linear extension であるとは, " if $v_p>v_q$, then $p>q$ $(p,q\in\{1,\cdots,d\})$ " を満たすことである. P が Γ に付随する poset の場合, Γ の linear extension と P の linear extension は一致する.

§4. d-Complete Poset の定義

この節では "d-complete poset" を定義する. "d-complete poset" という概念は R. A. Proctor によって定義された [13] poset である. しかし, これは我々の 一般化された GNW-algorithm には適さない. そこで, ここでは d-complete poset を graph として定義する. なお, graph としての d-complete poset の定義は仲田による.

まず, graph についての notation を用意する.

Definition 7. $x,y\in V$ とする.
x と y が:

1. $\phi(x)\cap\psi(y)=\varnothing$,
2. $\psi(x)\cap\phi(y)=\varnothing$, and
3. $x\neq y$,

を満たすとき $x|y$ と書く.

Definition 8. $v,u\in V$ とする.
v と u が:

1. $\#(\phi(v)\cap\psi(u))=1$, and
2. $\#(\psi(v)\cap\phi(u))=0$,

を満たすとき $v\to u$ と書く.

Definition 9. multi-di-graph $\Gamma = (V; A, \mathrm{o}, \mathrm{i})$ が *d-complete poset* であるとは,

(D0) $\#V < \infty$.

(D1) for each $v \in V$, we have $\#\phi(v) < \infty$.

(D2) for each $v \in V$, there exists no infinite v-paths.

(D3) for each $v, u \in V$, we have $\#(\phi(v) \cap \psi(u)) = 0, 1$.

(D4-a) Let $v, x, u \in V$ satisfy $v \to x \to u$ and $v|u$.
Then there exists a unique $y \in V$ such that $v \to y \to u$ and $x|y$.

(D4-b) Let $x, y, u \in V$ satisfy $x \to u \leftarrow y$ and $x|y$.
Then there exists a unique $v \in V$ such that $x \leftarrow v \to y$ and $v|u$.

(D4-c) Let $v, x, y \in V$ satisfy $x \leftarrow v \to y$ and $x|y$.
Then there exists a unique $u \in V$ such that $x \to u \leftarrow y$ and $v|u$, if it exists.

(D4-d) Let $(v; x_1, x_2, x_3) \in V^4$ satisfy

1. $x_i|x_j$ $(i \neq j)$, and
2. $v \to x_i$ or $v \leftarrow x_i$ $(i = 1, 2, 3)$.

Then we have

1. $v \to x_i$ $(i = 1, 2, 3)$.
2. for any $i, j = 1, 2, 3\,(i \neq j)$, there exists no $u \in V$ such that $x_i \to u \leftarrow x_j$ and $v|u$.

(D4-e) There exists no quartet $(v_1, v_2, v_3, v_4) \in V^4$ such that

1. $v_1|v_3$, $v_2|v_4$,
2. $v_1 \to v_2 \to v_3 \to v_4$, and $v_1 \to v_4$.

(D4-f) There exists no quartet $(v_1, v_2; u_1, u_2) \in V^4$ such that

1. $v_1|v_2$, $u_1|u_2$, and
2. $v_i \to u_j$ for $i, j = 1, 2$.

を満たすことである.

Remark 6. d-complete poset Γ に付随する poset が, 本来 Proctor が定義した "d-complete poset" である. ただし, Proctor は graph のことは意識せずに, 純粋に poset として (poset の言葉で) d-complete poset を定義した.

Remark 7. 条件 (D3) により, このグラフは simple である.
multi-arrow を持った graph で, 後述の Theorem 5.1 を満たすものについては [10].

§5. 主定理 と その系

Theorem 5.1 (Okamura [12] Nakada-Okamura [11]). Γ を *d-complete poset* とする. このとき, $\mathcal{B} \in \mathcal{L}(\Gamma)$ とすると, 一般化された *GNW-algorithm* は $\mathcal{B}$ を確率:

$$\mathrm{Prob}_\Gamma(\mathcal{B}) = \frac{\prod_{v \in V(\Gamma)}(1 + \#\phi(v))}{d!} \tag{5.1}$$

で生成する[4] .

Corollary 5.2. $\mathrm{Prob}_\Gamma()$ は $\mathcal{L}(\Gamma)$ 上の一様分布である.

Proof. (5.1) の右辺は linear extension に依存していないから. □

Corollary 5.3.

$$\#\mathcal{L}(\Gamma) = \frac{d!}{\prod_{v \in V(\Gamma)}(1 + \#\phi(v))}.$$

Proof. Corollary 5.2 から従う. □

Remark 8. simply-laced Kac-Moody Lie algebra のある特殊な条件を満たす integral weight に対しては, generalized Young diagram と呼ばれる, ある条件を満たす root の集合が定まり, これに対しては colored hook formula [8] という公式が成立する. Theorem 5.1 の証明は, まず, generalized Young diagram の言葉と, colored hook formula を応用することで証明される [11]. その後で, generalized Young diagram を graph の言葉で書き換える事によって (5.1) は得られる. なお, この書き換えができることは:

- すべての generalized Young diagram は, それが定める graph が Definition 9 を満たす (これは, [9] の結果を用いれば証明できる).
- すべての Definition 9 を満たす graph はある generalized Young diagram が定める graph と graph-同型である (これは, 川中の plain algorithm [7] を経由すれば証明できる).

から従う. 証明は本稿では省略する.

---

[4] Theorem 5.1 と同値な定理は最初, 岡村修志によって証明された [12].
[12] における証明は, R. A. Proctor による d-complete poset の分類定理 [13] に基づく case-by-case argument であり, 幾つかの場合は, 長時間の計算機による計算で示された.
一方, [11] では証明方法を大幅に改良し, 統一的な証明を与えている.

§6. 問題提起

問題1 (D0),(D1),(D2) を満たすグラフの中で, GNW-algorithm がすべての linear extension を定量的に等確率 (Theorem 5.1 の右辺) で生成するものをグラフの言葉のみで特徴づけよ (あるいは, d-complete poset より大きなクラスを見つけよ).

問題2 任意の poset P に対して (1),(2) を満たす acyclic multi-di-graph Γ は存在するか.

(1) Γ に付随する poset は P と順序同型.

(2) GNW-algorithm は Γ の linear extension を定量的に等確率に生成する.

- この問はおそらく negative に解かれるであろうが (#P-完全との関係のため), だとしても, 反例が知りたい.

References

[1] R. Bubley, and M.Dyer, *Faster random generation of linear extensions,* Discrete Mathematics, **201** (1999), 81-88.

[2] G. Brightwell, and P. Winkler, *Counting linear extensions,* Order **vol. 8, (No. 3)** (1991), 225-242.

[3] J. B. Carrell, *Vector fields, flag varieties, and Schubert calculus*, Proc. Hyderabad Conference on Algebraic Groups (ed. S. Ramanan), Manoj Prakashan, Madras, 1991.

[4] C. Greene, A. Nijenhuis, and H. S. Wilf, *A probabilistic proof of a formula for the number of Young tableaux of a given shape*, Adv. in Math. **31** (1979), 104-109.

[5] M. Huber, *Fast perfect sampling from Linear extensions,* Discrete Mathematics, **306** (2006), 420-428.

[6] A. Karzanov, and L. Khachiyan, *On the conductance of order Markov chains,* Order **vol. 8 (No. 1)** (1991), 7-15.

[7] N. Kawanaka, *Plain algorithm*, private talk.

[8] K. Nakada, *Colored hook formula for a generalized Young diagram*, Osaka J. of Math. **Vol. 54 No. 4** (2008), 1085-1120.

[9] K. Nakada, *q-Hook formula for a generalized Young diagram*, preprint.

[10] K. Nakada, *Another proof of hook formula for a shifted Young diagram*, in preparation.

[11] K. Nakada, and S. Okamura, *Uniform generation of standard tableaux of a generalized Young diagram*, preprint.

[12] S. Okamura, *An algorithm which generates a random standard tableau on a generalized Young diagram* (in Japanese), master's thesis, Osaka university, 2003.

[13] R. A. Proctor, *Dynkin diagram classification of λ-minuscule Bruhat lattices and of d-complete posets*, J. Algebraic Combin. **9** (1999), 61-94.

[14] B. E. Sagan, *On selecting a random shifted Young tableaux*, J. Algorithm **1** (1980), 213-234.